

Устройство автоматически ищет аудиофайлы всех форматов на устройстве, можно выбрать соответствующие файлы для добавления в плейлист. После добавления файлов можно прослушивать композиции. Для этого используются встроенные в приложение библиотеки аудиокодеков.

Заключение. Разработанное приложение предназначено для воспроизведения аудиофайлов на мобильных устройствах Android. При использовании разработанной системы появляется возможность легко находить аудиофайлы всех типов и добавлять их в разные плейлисты. На данный момент только одна программа на рынке мобильных приложений поддерживает настройку звука, но она является платной. Разработанное приложение является абсолютно бесплатным и позволяет качественно настраивать различные частоты звука.

Список цитируемых источников

1. *Дейтел, П.* Android для программистов: создаем приложения / П. Дейтел, Х. Дейтел, М. Моргано. — СПб. : Питер, 2013. — 560 с.
2. *Васильев, А. Н.* Java. Объектно-ориентированное программирование для магистров и бакалавров: базовый курс по объектно-ориентированному программированию / А. Н. Васильев. — СПб. : Питер, 2012. — 396 с.

УДК 004.89

В. О. Кандабаров, И. А. Камленок

Учреждение образования «Барановичский государственный университет», Барановичи

НЕЙРОННЫЕ СЕТИ В ВЕРСТКЕ САЙТОВ ПО КАРТИНКЕ

Введение. Основная цель развития искусственного интеллекта — нейронные сети. Одна из основных функций нейронных сетей — это моделирование человеческой нервной системы, способность к самообучению. В этом и есть главная особенность нейронной сети: они способны к самообучению на основе собственного опыта.

Основная часть. Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений, подобных тем, что делает человеческий мозг. Самые распространенные применения нейронных сетей: классификация — распределение данных по параметрам; предсказание — возможность предсказывать следующий шаг; распознавание — самое широкое применение нейронных сетей в настоящее время [1].

Построим и рассмотрим нейронную сеть в три итерации.

В первой версии сделаем минимальную версию, чтобы получить зависание движущихся частей. Вторая версия — HTML. В финальной версии Bootstrap создадим модель, которая может обобщать и исследовать слой LSTM (сети кротко срочной памяти).

Из данных нейронная сеть создает функции. Нейронная сеть создает функции для связывания входных данных с выходными данными. Построим веб-сайт с использованием нейронной сети с изображением “Hello World!”, а также сгенерируем разметку.

Нейронная сеть представляется в списках пиксельных значений. Существует этих значений от 0 до 255. Эти значения представляются в цветах — красном, зелёном, синем (рисунок 1).

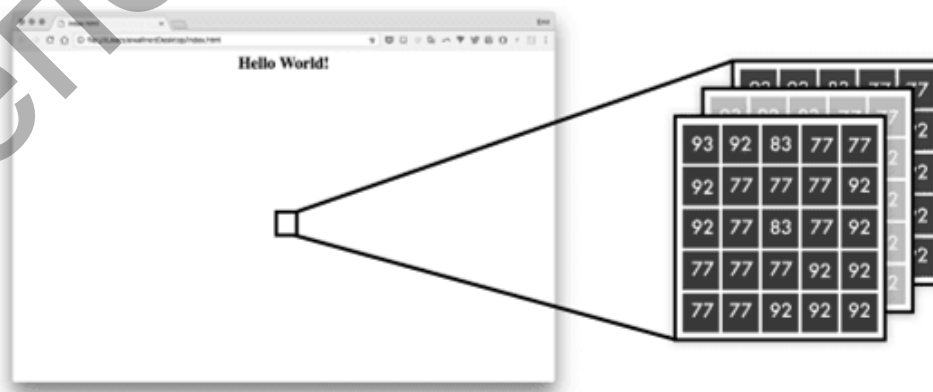


Рисунок 1 — Значения в цветах для нейронной сети

	start	I	can	code	end
vocabulary	0	0	0	0	0
start	1	0	0	0	0
I	0	1	0	0	0
can	0	0	1	0	0
code	0	0	0	1	0
end	0	0	0	0	1

Рисунок 2 — Закодированное приложение

	start	I	can	code	end
max sentence	10000	01000	00100	00010	00001
start	00000	00000	00000	00000	10000
start I	00000	00000	00000	10000	01000
start I can	00000	00000	10000	01000	00100
start I can code	00000	10000	01000	00100	00010
start I can code end	10000	01000	00100	00010	00001

Рисунок 3 — Заполнение словами



Рисунок 4 – Описание предсказаний

Чтобы представить разметку таким образом, что нейронная сеть понимает, используем одну горячую кодировку. Таким образом, предложение «Я могу закодировать» можно было бы отобразить так, как показано ниже (рисунок 2).

В приведенном выше графике мы включаем начальный и конечный теги. Эти теги являются сигналами, когда сеть начинает свои прогнозы и когда останавливается.

Для входных данных будем использовать предложения, начиная с первого слова, а затем добавляя каждое слово один за другим. Выходные данные — одно слово.

Предложения следуют той же логике, что и слова. Они также нуждаются в одной и той же входной длине. Вместо того чтобы ограничивать словарный запас, они связаны максимальной длиной предложения. Если она короче максимальной длины, вы заполняете ее пустыми словами, слово с нулями (рисунок 3).

Как видите, слова печатаются справа налево. Это заставляет каждое слово менять позицию для каждого тренировочного раунда. Это позволяет модели узнать последовательность, а не запоминать положение каждого слова.

В приведенном ниже графике есть четыре предсказания. Каждая строка представляет собой одно предсказание. Слева находятся изображения, представленные в трех цветовых каналах (красный, зеленый и синий), а также предыдущие слова. За пределами скобок указаны прогнозы один за другим, заканчивающиеся красным квадратом, чтобы отметить конец (рисунок 4) [2; 3].

Заключение. В приведенном примере работали с использованием трёх токенов: “start”, “Hello World!” и “end”. Задать токеном можно даже предложение.

Здесь был сделан прогноз:

- 10 эпох: start start start ;
- 100 эпох: start <HTML><center><H1>Hello World!</H1></center></HTML> <HTML><center><H1>Hello World!</H1></center></HTML> ;
- 300 эпох: start <HTML><center><H1>Hello World!</H1></center></HTML> end .

Список цитируемых источников

1. Зачем нужны нейронные сети [Электронный ресурс]. — Режим доступа: http://old.ci.ru/inform15_05/p_08.html . — Дата доступа: 17.02.2018.
2. Бум нейросетей: Кто делает нейронные сети, зачем они нужны и сколько денег могут приносить [Электронный ресурс]. — Режим доступа: <https://vc.ru/16843-neural-networks> . — Дата доступа: 17.02.2018.
3. Turning Design Mockups Into Code With Deep Learning [Электронный ресурс]. — Режим доступа: <https://blog.floydhub.com/Turning-design-mockups-into-code-with-deep-learning/> . — Дата доступа: 17.02.2018.